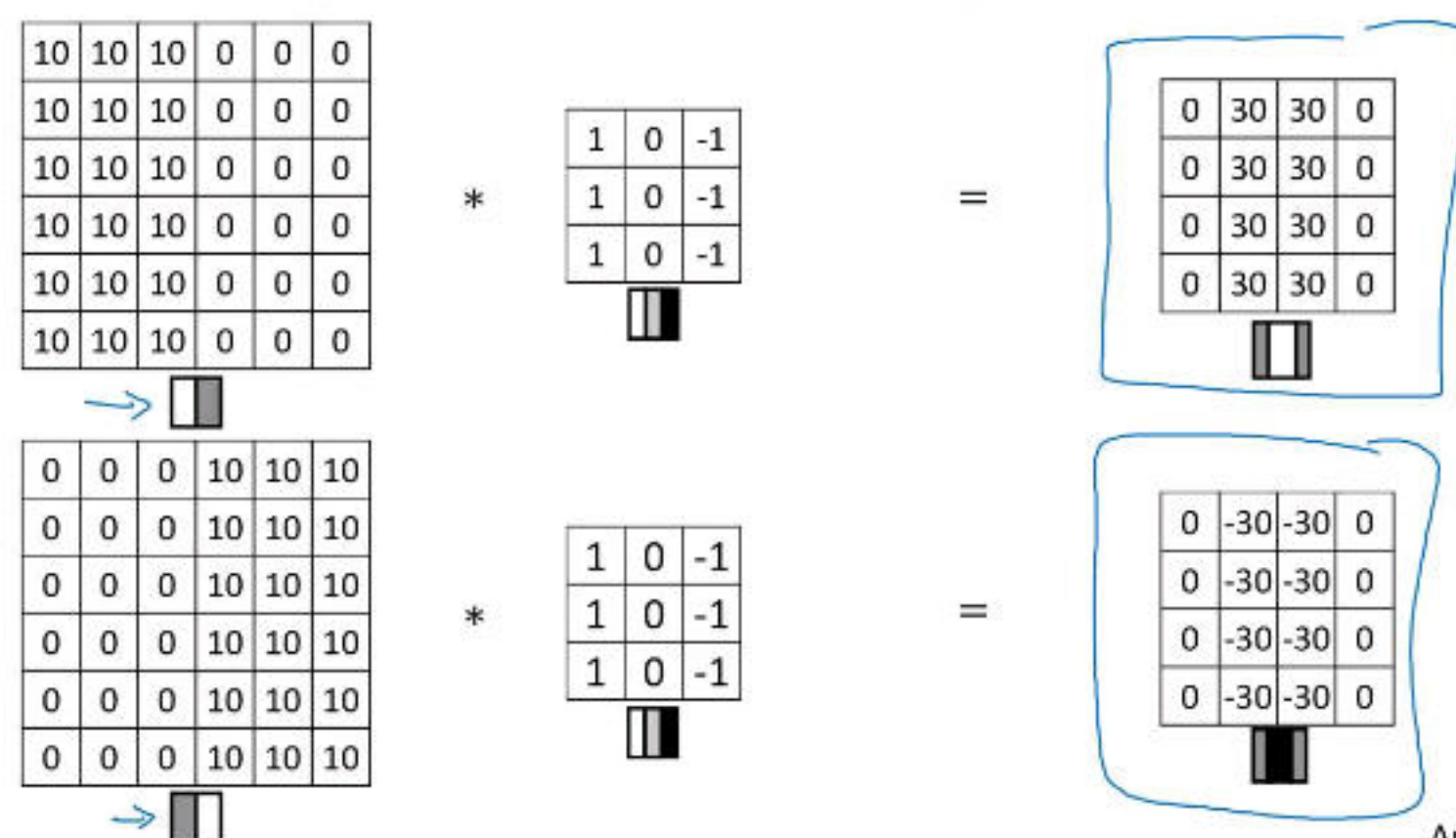


1 11

Deep learning.

Edge detection:

Vertical edge detection examples



Andrew Ng

Kernel 能检测出不同性质的垂直边缘。
水平的情况也是类似的。

Padding: $n \times n$ 的图像, $f \times f$ 的卷积核, padding 大小为 p .

卷积后图像大小: $n - f + 2p + 1$

$$\begin{cases} \text{Valid: } p = 0, \\ \text{Same: } n - f + 2p + 1 = n \Rightarrow p = \frac{f-1}{2}. \end{cases}$$

$$\text{Stride} = S, \text{ Output size} = \left\lfloor \frac{n + 2p - f}{S} \right\rfloor + 1$$

卷积与数学上的运算是有所差别的, “惯例名称”

n_c 通道做卷积, n_k 层卷积核, 输出一个矩阵。

多个卷积核则输出多个矩阵。

CNN 通常最后一层是 Logistic / SoftMax 层。

1 11

Pooling Layer.

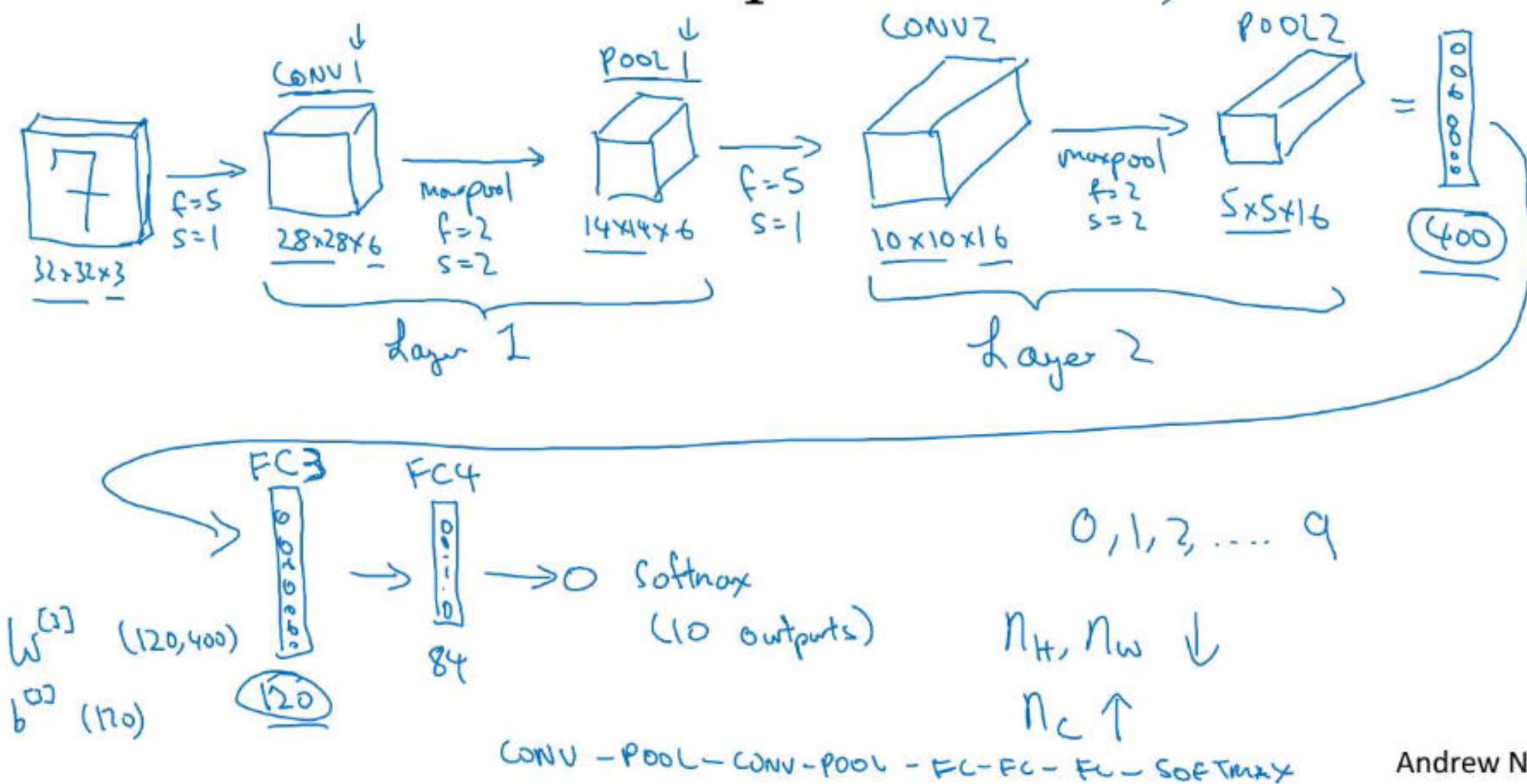
2个超参, f, s , 常用: $f=2, s=2$; $f=3, s=2$.

$\left\{ \begin{array}{l} \text{Max Pooling} \\ \text{Average Pooling} \end{array} \right.$

对每个 channel 都做.

典型 CNN

Neural network example (LeNet-5)



size \downarrow , 一个 w, b 算一层.

卷积的优势: $\left\{ \begin{array}{l} \text{参数共享: 特征提取复用} \\ \text{稀疏连接:} \end{array} \right.$

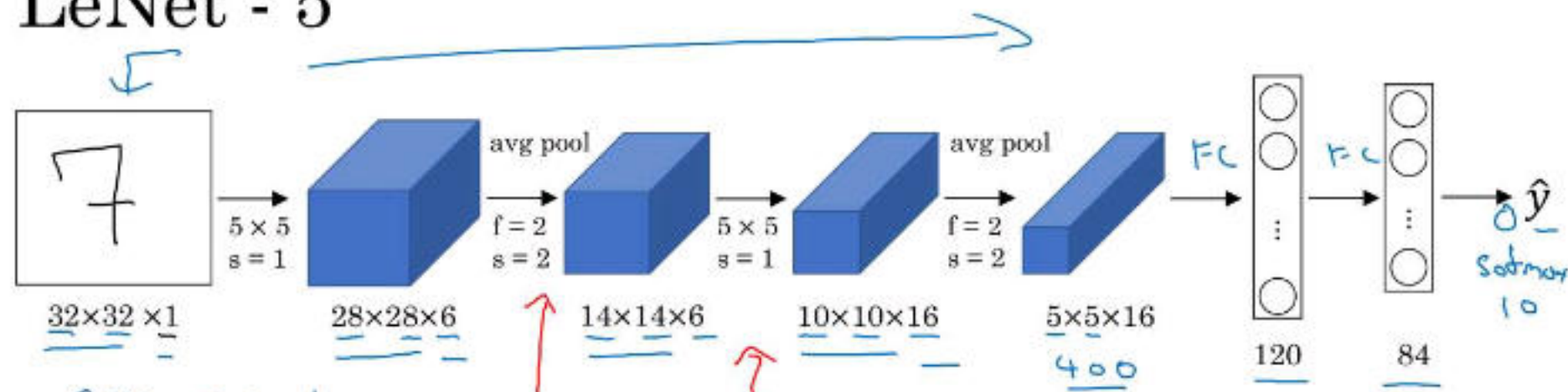
translation invariance.

1

12

Classic CNN.

LeNet - 5



60k parameters.

$n_H, n_w \downarrow$ $n_c \uparrow$ non-linearly after pooling $n_H \times n_w \times n_c$ $f \times f \times n_c$

conv pool conv pool fc fc output

Activat: Sigmoid/tanh ReLU

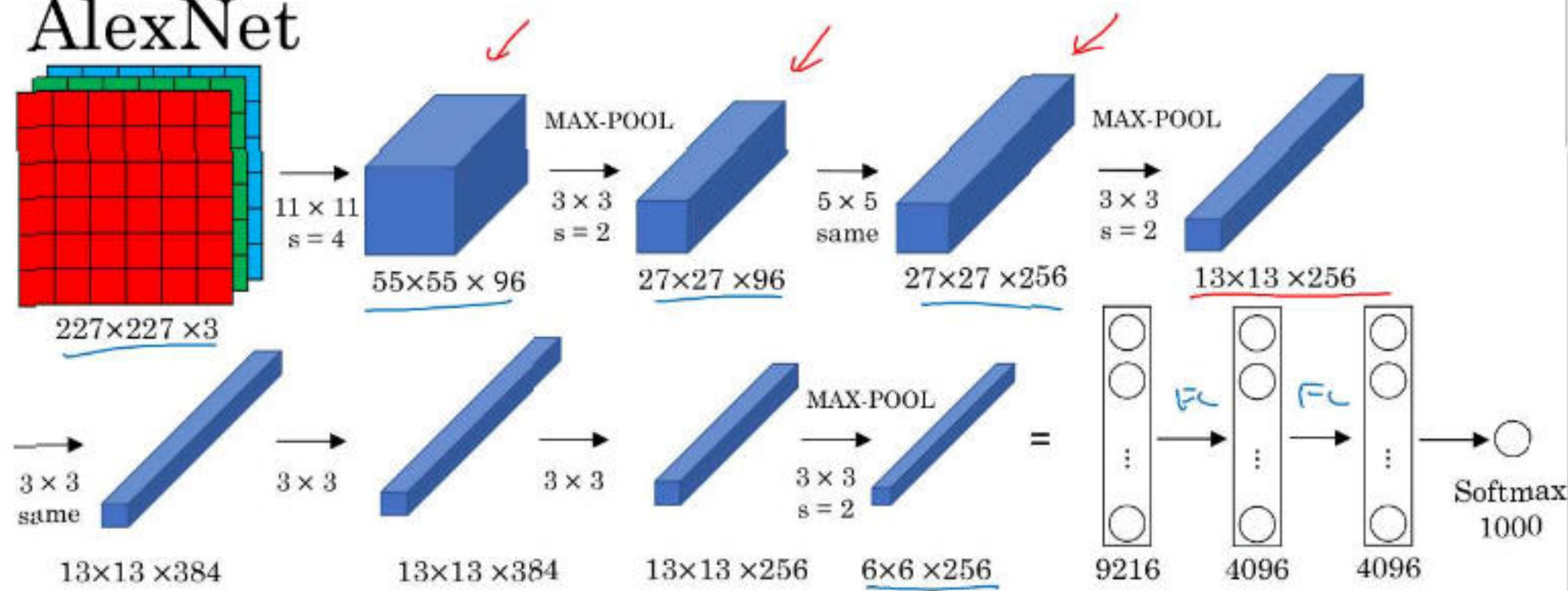
II III

[LeCun et al., 1998. Gradient-based learning applied to document recognition]

Andrew Ng

$n_H, n_w \downarrow$. 用 Sigmoid / tanh 激活, Pool 也加了激活.
参数规模: 60k

AlexNet



- Similar to LeNet, but much bigger.

- ReLU

- Multiple GPUs.

- Local Response Normalization (LRN)

13 13 256

160M parameters

[Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks]

Andrew Ng

Act: ReLU, 参数规模: 60M.
filter 数量大大增加.

1 12

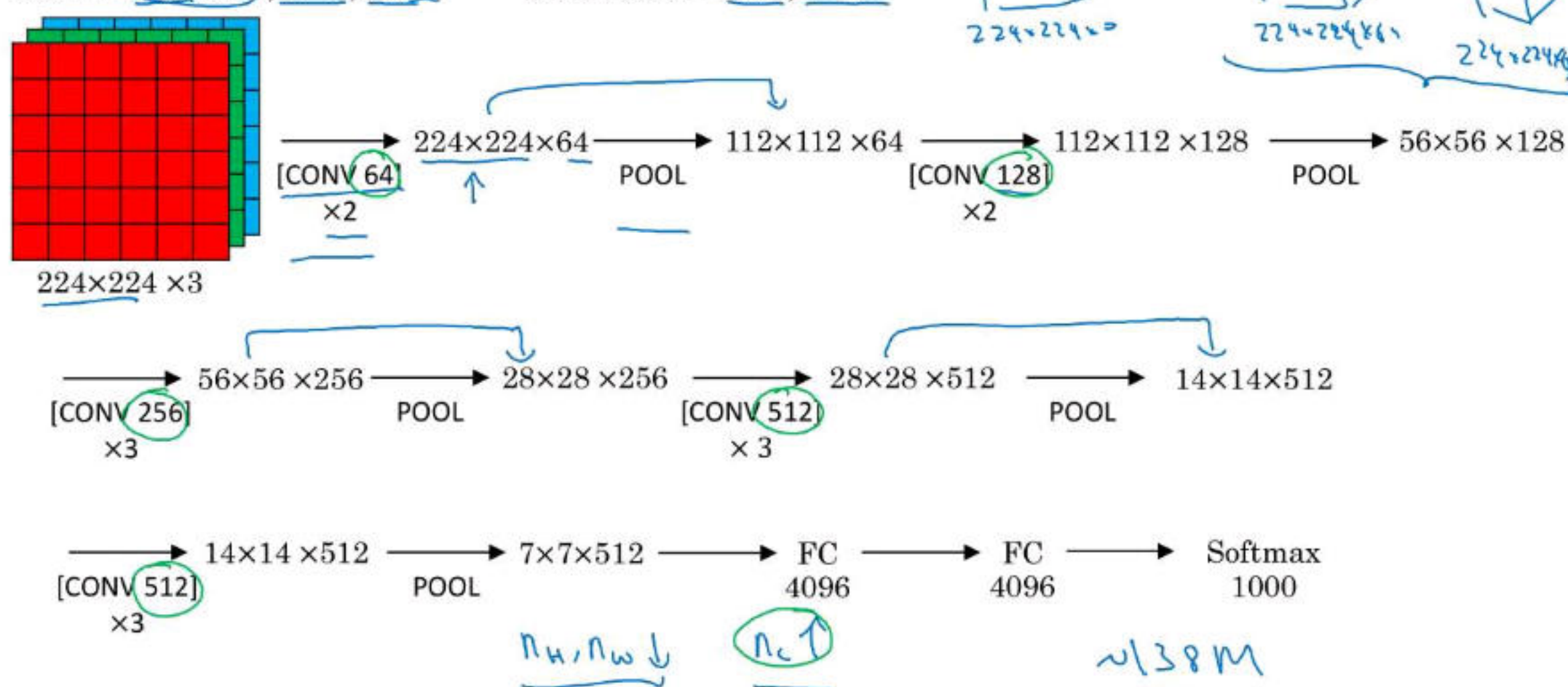
VGG - 16

CONV = 3x3 filter, s = 1, same

MAX-POOL = 2x2, s = 2

M-Pencil

100%



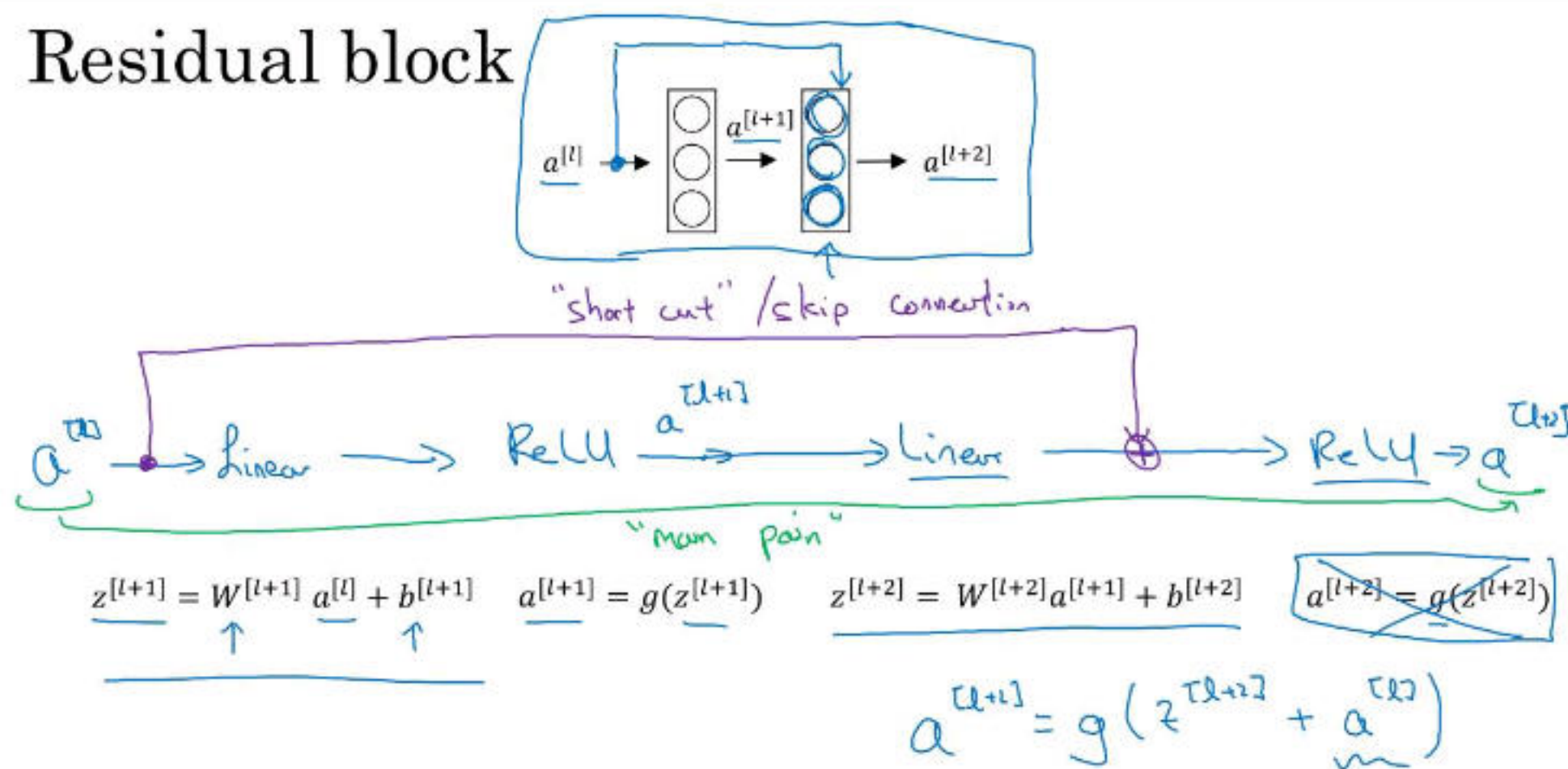
[Simonyan & Zisserman 2015. Very deep convolutional networks for large-scale image recognition]

Andrew Ng

$s=2$, size 减半. # filter 加倍. 1000 类 SoftMax.
参数规模: 138 M.

残差网络.

Residual block



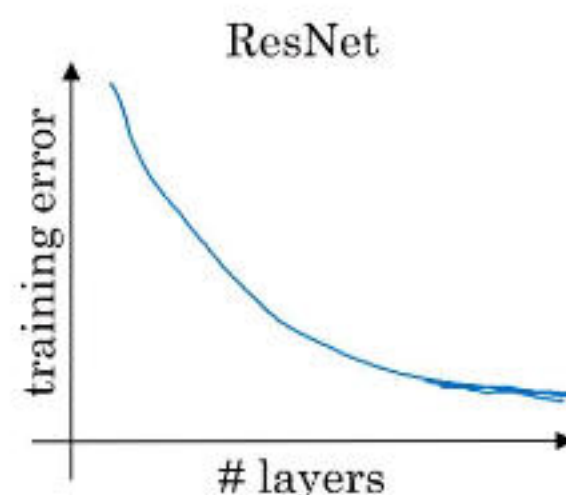
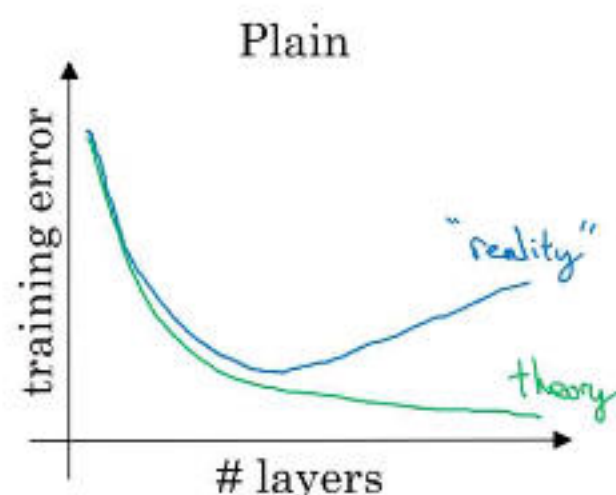
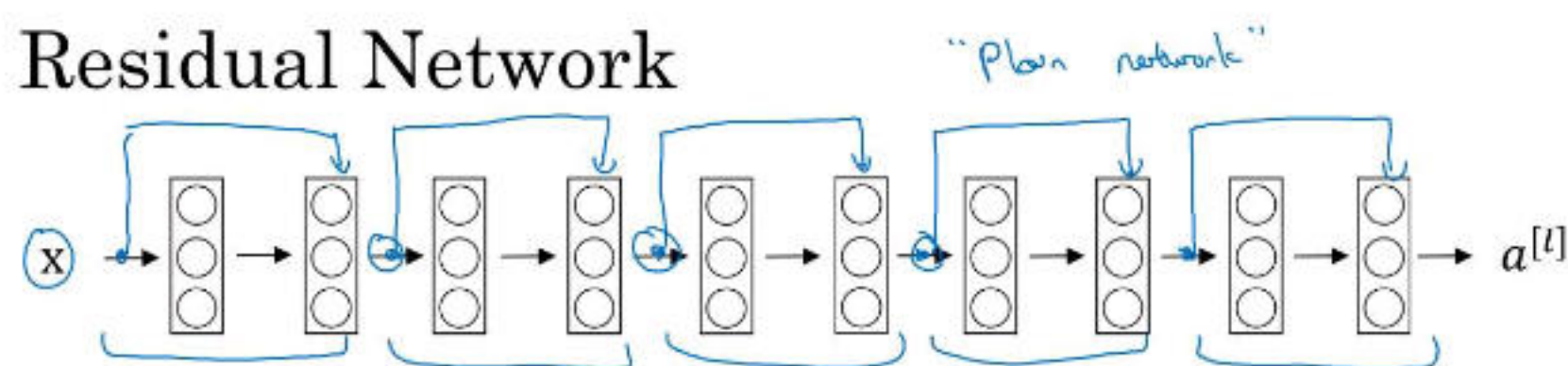
[He et al., 2015. Deep residual networks for image recognition]

Andrew Ng

$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]}).$$

"Short-Cut".

Residual Network



[He et al., 2015. Deep residual networks for image recognition]

Andrew Ng

在训练集上的 Error 即便 depth ↑, 也是 ↓.

为什么好用?

1): Reg 后, $a^{[L+1]} = g(z^{[L+1]} + a^{[1]})$ (Relu).

$$= g\left(\underbrace{w^{[L+1]} \cdot a^{[L]}}_{0} + \underbrace{b^{[L+1]}}_{0} + a^{[1]}\right) = a^{[1]}$$

学会了 1 关系.

2): 若 $a^{[L+1]}$ 与 $a^{[1]}$ 维度不同, $w_s a^{[1]}$ 调整维度. w_s 可以 0 pad $a^{[1]}$, 也可自学习.

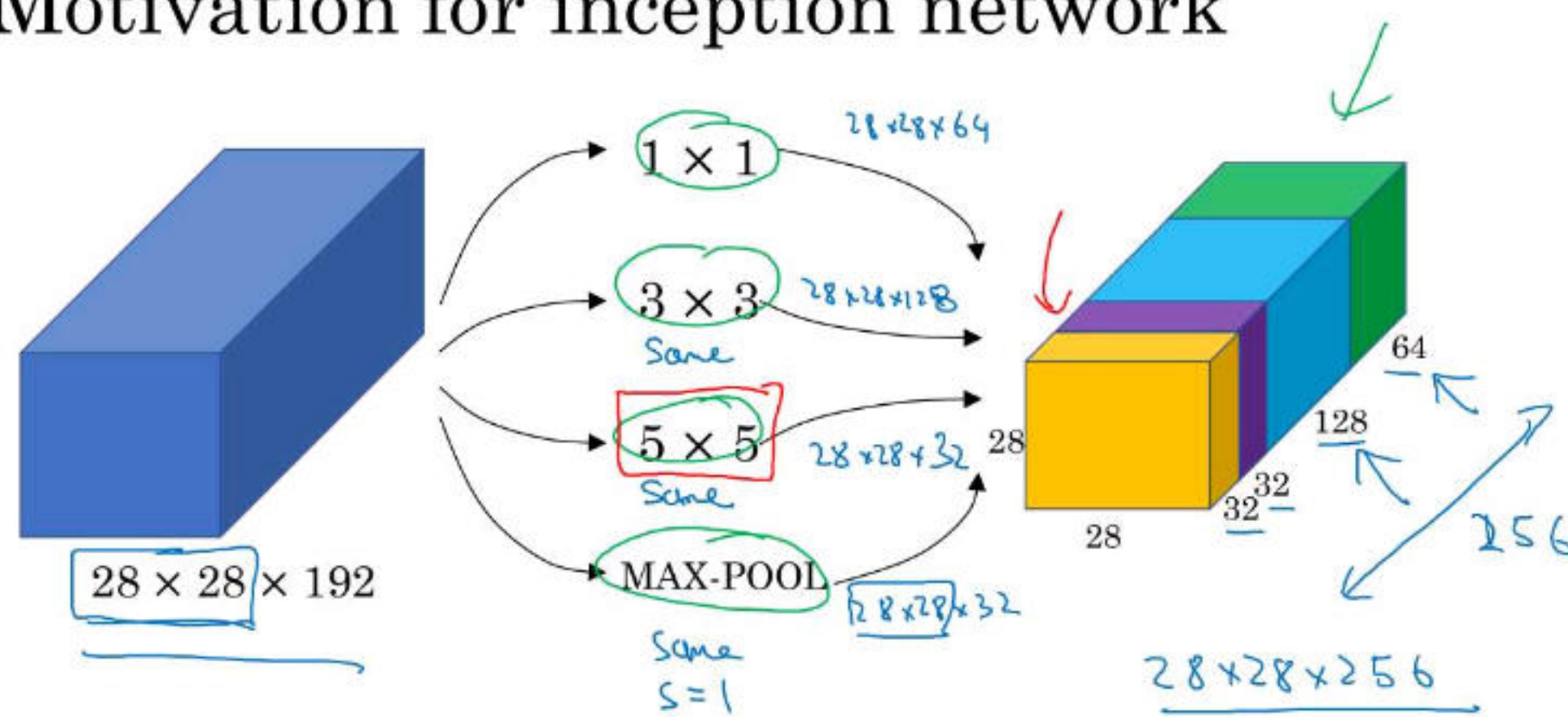
1 × 1 卷积.

类似于:



相当于一层 FC.
网络中的网络.

Motivation for inception network



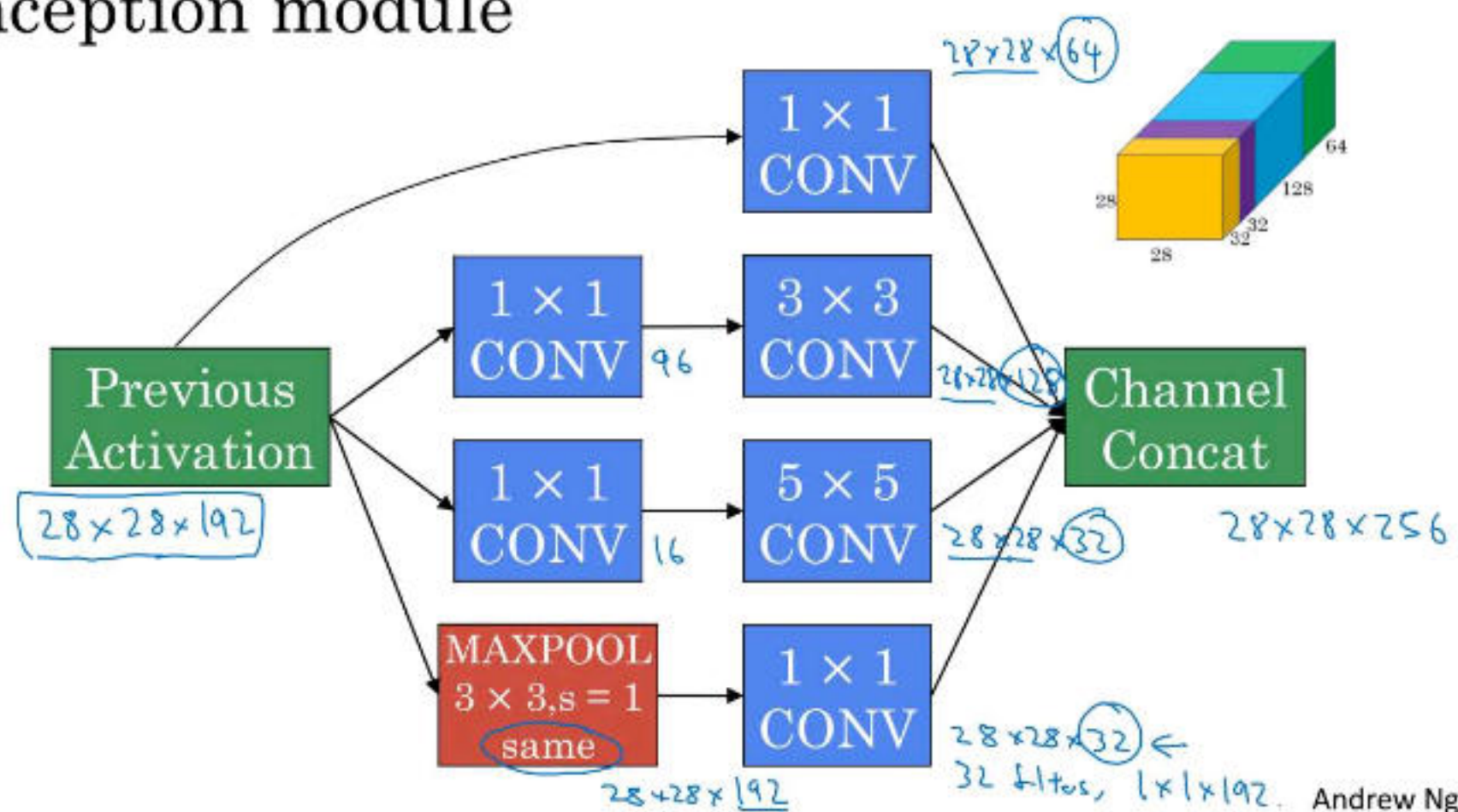
[Szegedy et al. 2014. Going deeper with convolutions]

Andrew Ng

inception 网络集成了各种 filter, 自动选择使用哪一个 filter.
缺点: 计算代价太高.

利用 1x1 filter 可以降低 Computational Cost.

Inception module



Andrew Ng

Inception 网络由基本的 module 变形、堆叠而成.

1 12

Transfer Learning.

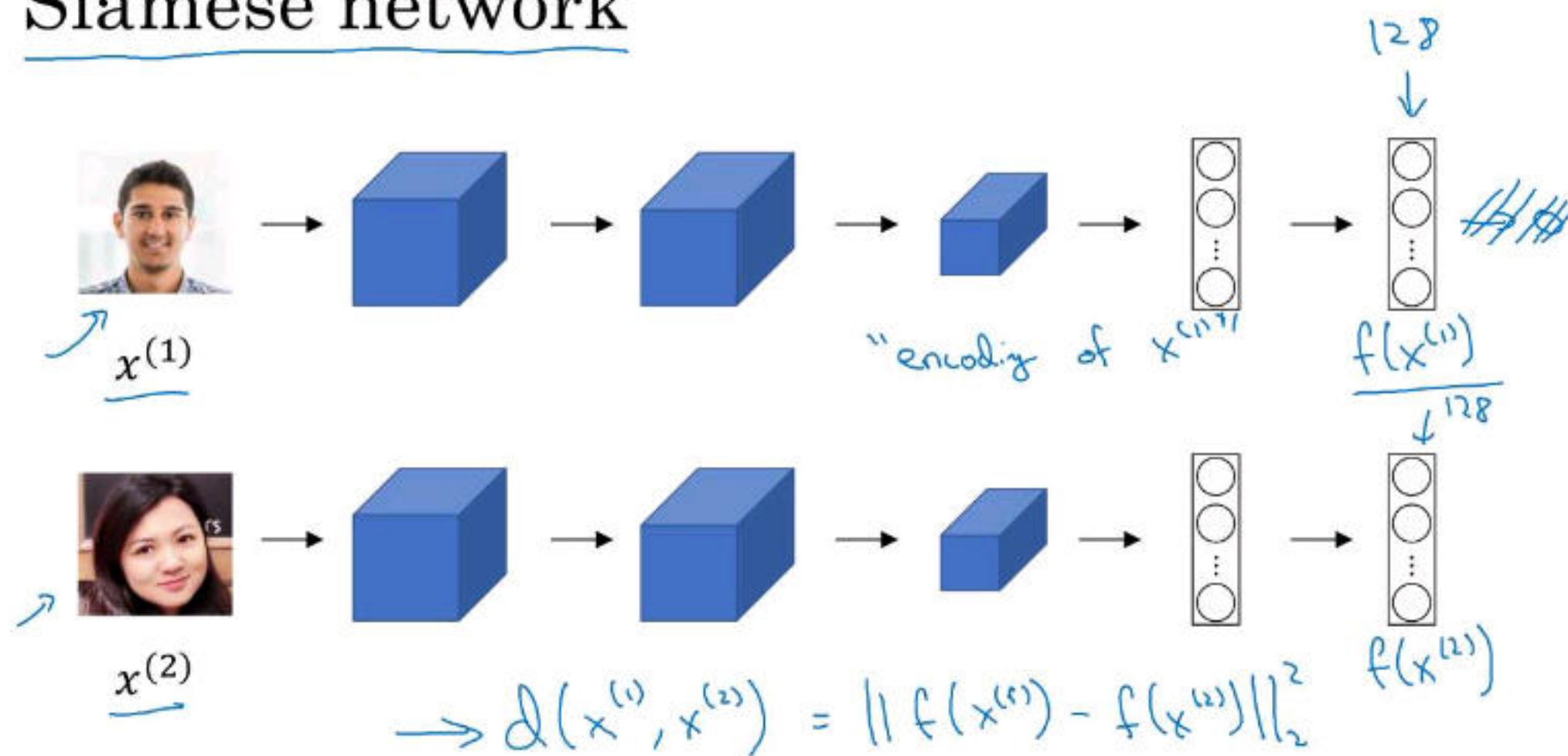
- 只训练 SoftMax (少量 Data Set).
- 训练最后几层 (中等 Data Set).
- 重新训练 (用预训练参数 Initialize). (大量一).

Data Augmentation.

- 图像变换: Mirror, Shearing ...
- 色彩变换: Color Shifting.

人脸识别的 CNN

Siamese network



[Taigman et. al., 2014. DeepFace closing the gap to human level performance]

Andrew Ng

是一种 Encoder. 用于 d 的计算.

1 13

Loss function

Given 3 image A, P, N :

$$L(A, P, N) = \max(\underbrace{\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha}_{> 0}, 0)$$

$$J = \sum_{i=1}^m L(A^{(i)}, P^{(i)}, N^{(i)})$$

A, P
↑ ↑

Training set: 10k pictures of 1k persons

[Schroff et al., 2015, FaceNet: A unified embedding for face recognition and clustering]

Andrew Ng

Siamese 的 Loss.

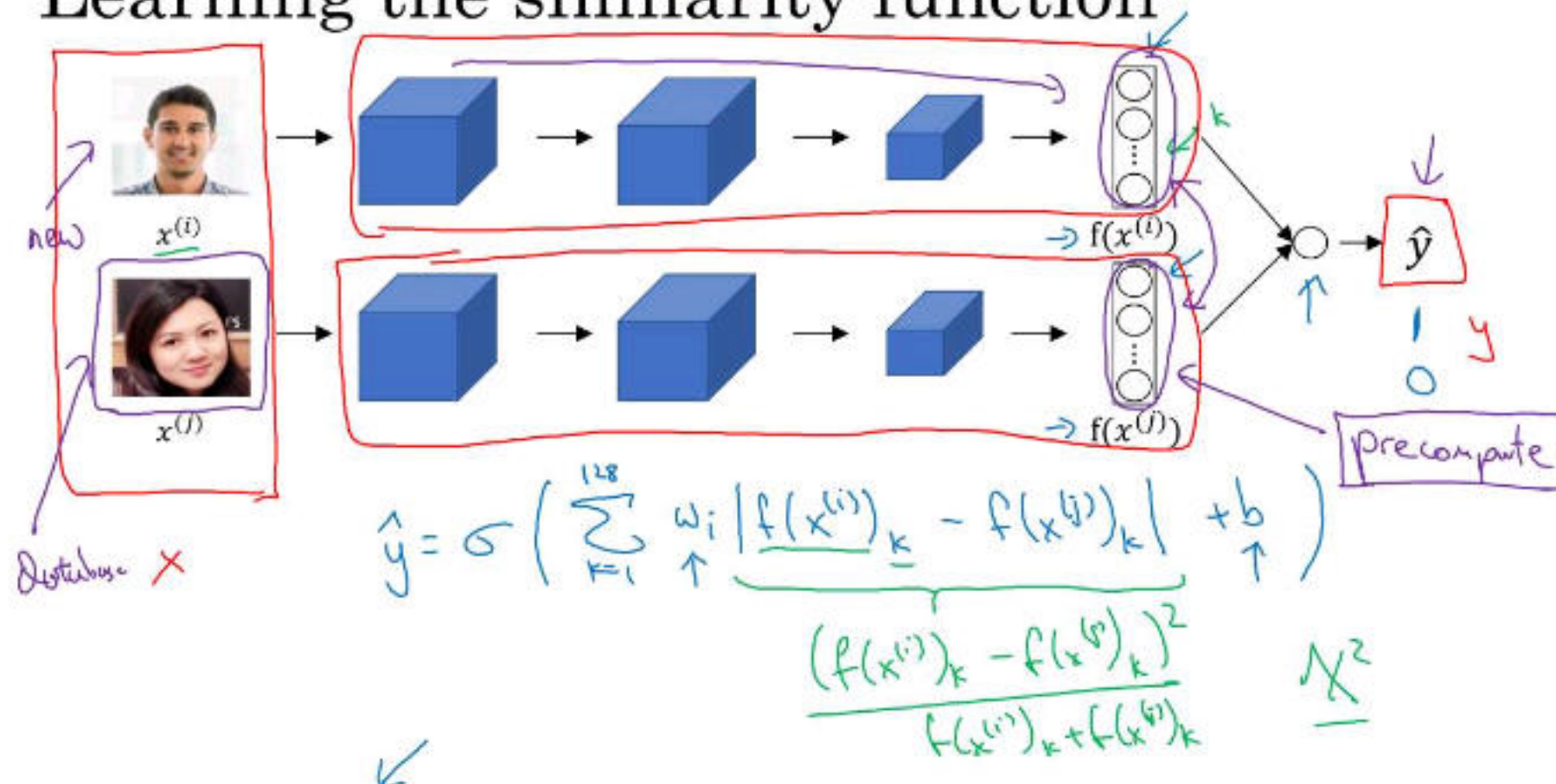
$$L(A, P, N) = \max\{\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0\}.$$

$$\|f(A) - f(P)\|^2 + \alpha \leq \|f(A) - f(N)\|^2.$$

$$J(\theta) = \sum_{i=1}^m L(A^{(i)}, P^{(i)}, N^{(i)}).$$

人脸识别的二分类.

Learning the similarity function



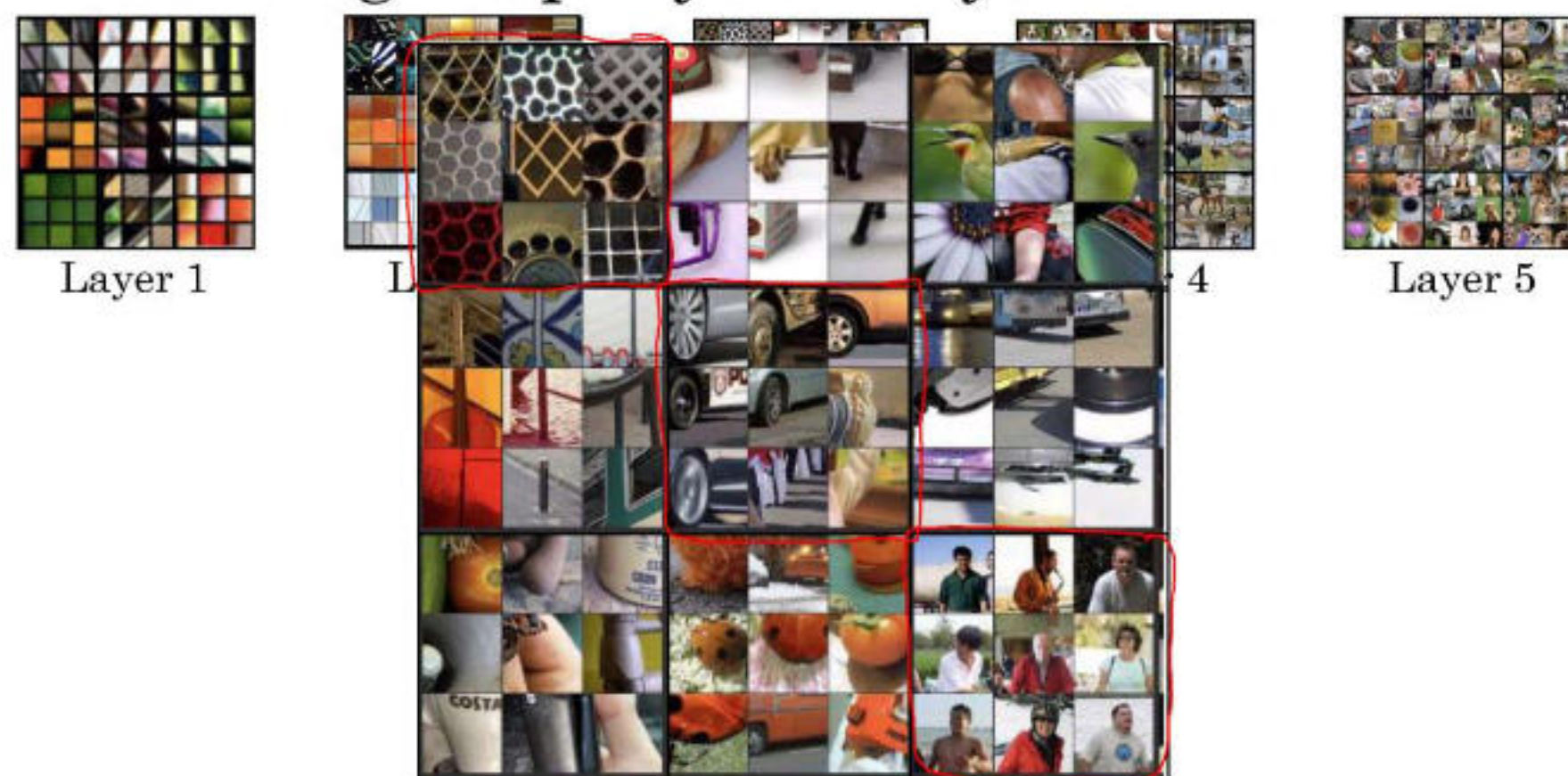
[Taigman et. al., 2014, DeepFace closing the gap to human level performance]

Andrew Ng

多一层 Logistic 回归, 训练一个分类器.

Neural Transfer.

Visualizing deep layers: Layer 3



Andrew Ng



卷积次数↑, 提取的模式与特征越复杂.

Content cost function

M-Pencil

100%

Content: C Style: S Generation: G

$$J(G) = \alpha J_{\text{content}}(C, G) + \beta J_{\text{style}}(S, G)$$

- Say you use hidden layer l to compute content cost.
- Use pre-trained ConvNet. (E.g., VGG network)
- Let $a^{[l](C)}$ and $a^{[l](G)}$ be the activation of layer l on the images
- If $a^{[l](C)}$ and $a^{[l](G)}$ are similar, both images have similar content

$$J_{\text{content}}(C, G) = \frac{1}{2} \| \underbrace{a^{[l](C)}}_{\text{activation of layer } l \text{ on } C} - \underbrace{a^{[l](G)}}_{\text{activation of layer } l \text{ on } G} \|^2$$

[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng

$$J_{\text{content}} = \| a^{[l](C)} - a^{[l](G)} \|_F^2$$

Style matrix

Let $a_{i,j,k}^{[l]}$ = activation at (i,j,k) . $G^{[l]}$ is $n_c^{[l]} \times n_c^{[l]}$

$\rightarrow G_{kk'}^{[l](s)} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a_{ijk}^{[l](s)} a_{ijk'}^{[l](s)}$
 $\rightarrow G_{kk'}^{[l](G)} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a_{ijk}^{[l](G)} a_{ijk'}^{[l](G)}$

"Gram matrix"

n_c
 $G_{kk'}^{[l]}$
 $k=1, \dots, n_c$

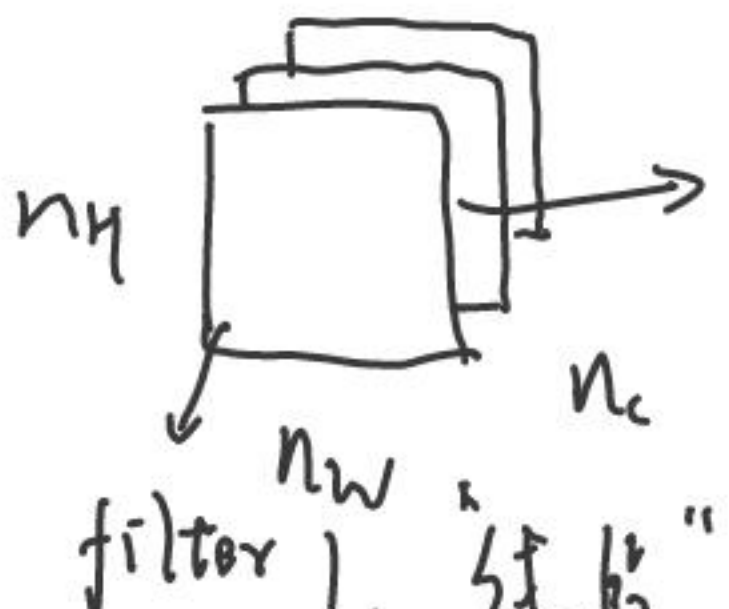
$$J_{\text{style}}^{[l]}(S, G) = \frac{1}{\binom{n_H^{[l]} n_W^{[l]} n_c^{[l]}}{2}} \|G^{[l](s)} - G^{[l](G)}\|_F^2$$

$$= \frac{1}{(2n_H^{[l]} n_W^{[l]} n_c^{[l]})^2} \sum_k \sum_{k'} (G_{kk'}^{[l](s)} - G_{kk'}^{[l](G)})^2$$

\uparrow
 β

[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng



filter 1. 红 (red). filter 2. Color. Style 体现为不同特征组合.

$$G_{kk'}^{[l]} = \sum_i \sum_j a_{ijk}^{[l]} a_{ijk'}^{[l]}$$

(展开为向量的内积)

$$J_{\text{style}}^{[l]}(S, G) = \frac{1}{k} \cdot \|G^{[l](s)} - G^{[l](G)}\|_F^2$$

$$k = (2n_W^{[l]} n_H^{[l]} n_c^{[l]})^2$$

$$J = \alpha \cdot J_{\text{content}} + \beta \cdot J_{\text{style}}$$

$$G = G - \alpha \cdot \frac{\partial J}{\partial G}$$